Multivariate Statistical Tests for Comparing Classification Algorithms

Olcay Taner Yıldız[1], Özlem Aslan[2], and Ethem Alpaydın[2]

[1] Dept. of Computer Engineering, Işık University, TR-34980, Istanbul, Turkey
[2] Dept. of Computer Engineering, Boğaziçi University, TR-34342, Istanbul, Turkey

Abstract. The misclassification error which is usually used in tests to compare classification algorithms, does not make a distinction between the sources of error, namely, false positives and false negatives. Instead of summing these in a single number, we propose to collect multivariate statistics and use multivariate tests on them. Information retrieval uses the measures of precision and recall, and signal detection uses true positive rate (tpr) and false positive rate (fpr) and a multivariate test can also use such two values instead of combining them in a single value, such as error or average precision. For example, we can have bivariate tests for (precision, recall) or (tpr, fpr). We propose to use the pairwise test based on Hotelling's multivariate T^2 test to compare two algorithms or multivariate analysis of variance (MANOVA) to compare $L > 2$ algorithms. In our experiments, we show that the multivariate tests have higher power than the univariate error test, that is, they can detect differences that the error test cannot, and we also discuss how the decisions made by different multivariate tests differ, to be able to point out where to use which. We also show how multivariate or univariate pairwise tests can be used as post-hoc tests after MANOVA to find cliques of algorithms, or order them along separate dimensions.

1 Introduction

For a typical machine learning application, there are multiple candidate algorithms and we need to choose one among many. In supervised learning, this is typically done by comparing errors, and in classification with two classes, the misclassification error is the sum of false positives and false negatives (see Table 1(a)). However, misclassification error does not make a distinction between false positives and false negatives, and various other measures have been proposed depending on the type of error we focus on (see Table 1(b)). In information retrieval, the two measures used are precision and recall, and in signal detection, they are true positive rate (tpr) and false positive rate (fpr). People also use curves of these or areas under such curves. These different set of measures have different uses, as we will discuss later.

In comparing classification algorithms, we use statistical tests to make sure that the difference is *significant,* that is, big enough that it could not have happened by chance, or in other words, very unlikely to have been caused by

C.A. Coello Coello (Ed.): LION 5, LNCS 6683, pp. 1–15, 2011.

Table 1. (a) 2×2 confusion matrix for two classes. (b) Different performance measures.

(a)

	Predicted class		
True class	Positive	Negative	Sum
Positive	*tp*	*fn*	*p*
Negative	*fp*	*tn*	*n*
Sum	*p'*	*n'*	

(b)

Name	Formula
error	$(fp+fn)/(p+n)$
accuracy	$(tp+tn)/(p+n)$
tpr	tp/p
fpr	fp/n
precision	tp/p'
recall	tp/p

chance – the so-called *p-value* of the test. To be able to measure the effect of chance (e.g., variance due to small changes in the training set), typically, one does training and validation a number of times, possibly by resampling using cross-validation. For example, with k training and validation dataset pairs, we train the classification algorithms on the k training sets and obtain the k confusion matrices on the validation sets. From these, we can for example calculate the k misclassification error values and to compare two algorithms, we can use a pairwise statistical test [1] to see whether the two algorithms lead to classifiers with equal expected error. When there are more than two to compare, one can use analysis of variance (ANOVA) to check if all have equal expected error. It is critical that such tests are *paired,* that is, we use the same training and validation data with all algorithms so that whatever difference we observe is due to the algorithm, and not due to any randomness in resampling the data.

We note the disadvantage of using error here; such tests cannot make a distinction between false positives and false negatives. Two classifiers may have the same error but one may have all its error due to false positives, the other all due to false negatives, and we will not be able to detect this difference if our comparison metric is simply the error; see Figure 1 for an example.

In this paper, we propose *multivariate tests* that can do comparison using multiple measures and not just a single one, i.e., error. That is, from the k confusion matrices, we will collect *multivariate statistics* such as a two-dimensional vector of (tpr, fpr) or (precision, recall), and do a bivariate test. We can also do a four-variate test using the whole 2×2 confusion matrix or any other vector of measurements. Statistical tests in the machine learning literature are all univariate; to the best of our knowledge, our use of multivariate tests in performance comparison of machine learning algorithms is the first.

The need to combine different measures have been noticed before. Average precision combines precision and recall, for example, Caruana et al. (2004) [2] compared different performance metrics such as accuracy, lift, F-Score, area under the ROC curve, average precision, precision/recall break-even point, squared error, cross entropy, and probability calibration; they showed that these metrics are correlated and proposed a new measure SAR as the average of Squared error, Accuracy and Roc area. Seliya et al. (2009) [3] calculated different measures too and going one step further proposed to combine them taking the correlation into

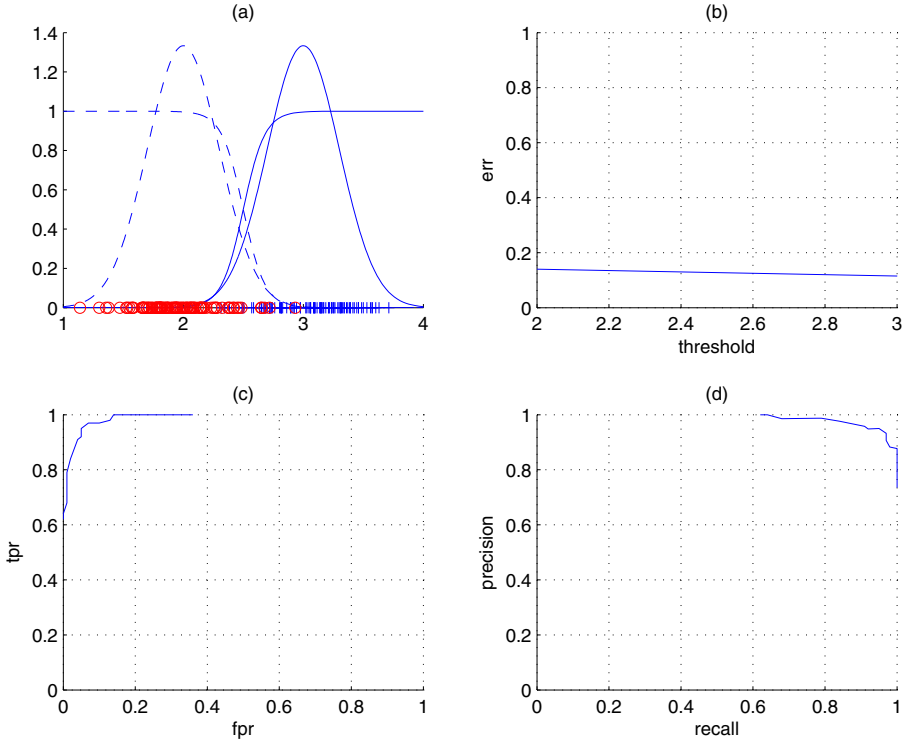


Fig. 1. Example showing that error is not the best measure in comparison. The negative and positive instances are normally distributed with their means at 2 and 3 respectively; both have standard deviation 0.3. In (a), we see the two densities, the posterior probabilities and 100 instances sampled from each. We have a classifier that chooses the positive class if the input is greater than a threshold (corresponding to a threshold on the posterior of the positive class) and what we then do, is move this threshold of decision gradually from 2 to 3 (corresponding to increasing the posterior threshold from 0 to 1). As we see in (b), the error does not change; the number of false positives decreases but the number of false negatives increase in equal amount. In (c) and (d), we see that if we use (tpr, fpr) and (precision, recall) as measures of performance, the values differ as the threshold is changed. As the threshold increases, the number of true positives decrease which decreases tpr and recall; but because false positives decrease, fpr decreases and precision increases. Note that in (c) and (d), as we increase the threshold, we move from the right to the left along the curves. (Tpr, fpr) and (precision, recall) can detect a difference due to different thresholds because they make a distinction between false positives and false negatives. For example, if we had two classifiers one with threshold at 2 and another with threshold at 3, a pairwise test on error would not be able to detect any difference between them, but tests on (tpr, fpr) or (precision, recall) would. The aim of this paper is the discussion of such tests.

account. Note however that these are for reporting performances only and they include no statistical methodology for testing or comparison, as we do here.

This paper is organized as follows: To compare two algorithms, we discuss the pairwise univariate test and the proposed multivariate test in Section 2. When there are $L > 2$ algorithms to compare, we can use univariate and multivariate ANOVA, as discussed in Section 3. We give our experimental results in Section 4 and conclude in Section 5.

2 Pairwise Comparison

Let us say we have two classification algorithms. We train and validate the two algorithms on k training/validation data folds and calculate the resulting k separate 2×2 confusion matrices $M_{ij}, i = 1, 2, j = 1, \ldots, k$, on the validation sets in the same format as shown in Table 1(a).

2.1 Univariate Case

If we want to compare in terms of error, for both algorithms and all k folds, we calculate $e_{ij} = fp_{ij} + fn_{ij}$ and then the paired difference between the errors

$$d_j = e_{1j} - e_{2j}$$

and we test if these differences come from a population with zero mean:

$$H_0 : \mu_d = 0 \text{ vs. } H_1 : \mu_d \neq 0$$

For the *univariate paired t test*, we calculate the average and the standard deviation:

$$\overline{d} = \sum_{j=1}^{k} d_j / k \ , \ s_d = \frac{\sum_j (d_j - \overline{d})^2}{k - 1}$$

Under the null hypothesis that the two algorithms have the same expected error, we know that

$$t' = \sqrt{k} \frac{\overline{d}}{s_d} \tag{1}$$

is t distributed with $k - 1$ degrees of freedom. We reject H_0 if $|t'| > t_{\alpha/2,k-1}$ with $(1 - \alpha)100$ % confidence.

2.2 Multivariate Case

If we do not want to reduce to a single statistic and want to use a set of values in comparison, we need a test that can use vectors instead of scalars. In such a case, we want to compare the means of two p-dimensional populations, that is, we want to test for the null hypothesis $H_0 : \boldsymbol{\mu}_1 - \boldsymbol{\mu}_2 = \mathbf{0}$. If we want to compare in terms of (tpr, fpr) or (precision, recall), then $p = 2$. Note that using the same setting, it is also possible to define a multivariate test on (sensitivity, specificity), or consider all four entries in the confusion matrix, in which case

$p = 4$. As before, we train and validate both algorithms with the same folds and use a paired test, except that now the test is multivariate.

Let us say $\boldsymbol{x}_{ij} \in \Re^p$ is the performance vector containing p performance values. For the *multivariate paired Hotelling's test,* we calculate the paired difference vectors

$$\boldsymbol{d}_j = \boldsymbol{x}_{1j} - \boldsymbol{x}_{2j}$$

and check if they come from a p-variate Gaussian with zero mean:

$$H_0 : \boldsymbol{\mu}_d = \mathbf{0} \text{ vs. } H_1 : \boldsymbol{\mu}_d \neq \mathbf{0}$$

We calculate the average vector and the covariance matrix:

$$\overline{\boldsymbol{d}} = \sum_{j=1}^{k} \boldsymbol{d}_j / k \ , \ \mathbf{S}_d = \frac{1}{k-1} \sum_j (\boldsymbol{d}_j - \overline{\boldsymbol{d}})(\boldsymbol{d}_j - \overline{\boldsymbol{d}})^T$$

Under the null hypothesis that the two algorithms have the same expected behavior, we know that [4]

$$T'^2 = k \overline{\boldsymbol{d}}^T \mathbf{S}_d^{-1} \overline{\boldsymbol{d}} \tag{2}$$

is *Hotelling's* T^2 distributed with p and $k-1$ degrees of freedom. We reject the null hypothesis if $T'^2 > T^2_{\alpha,p,k-1}$. Hotelling's $T^2(p, m)$ can be approximated using F distribution via the formula

$$\left(\frac{m-p+1}{mp} \right) T^2_{p,m} \sim F_{m,m-p+1} \tag{3}$$

Note that we calculate our measures such as tpr, precision, and so on, from entries in the 2×2 confusion matrix; these are counts of indicator random variables (they are 0/1 Bernoulli random variables) caused by the same event (the trained classifier) and the total counts are then dependent binomial random variables. We know from the central limit theorem that the binomial converges to the Gaussian unless the sample (here, the validation set size) is very small and hence the assumption of joint multivariate normality makes sense. Remember that all parametric tests based on error also use the same assumption.

When $p = 1$, this multivariate test reduces to the univariate t test of Section 2.1. Just like $\overline{d}/s_d$ of (1) measuring the normalized distance in one dimension, $\overline{\boldsymbol{d}}^T \mathbf{S}_d^{-1} \overline{\boldsymbol{d}}$ of (2) measures the (squared) normalized distance in p dimensions.

If the multivariate test rejects, we can do p *post-hoc* univariate tests to check which one(s) of the variates cause(s) a rejection. For example, if a multivariate test on (precision, recall) rejects, we may want to check if the difference is due to a significant difference in precision, recall, or both. For testing difference in variate l, we use the univariate test in (1) and calculate

$$t'_l = \sqrt{k} \frac{\overline{d}_l}{\mathbf{S}_{d,ll}} \tag{4}$$

and reject $H_0 : \mu_{d,l} = 0$ if $|t'_l| > t_{\alpha/2,k-1}$.

Note that it may be the case that none of the univariate differences is significant whereas the multivariate one is, and the linear combination of variates that cause the maximum difference can be calculated as

$$\boldsymbol{w} = \mathbf{S}_d^{-1}\overline{\boldsymbol{d}} \tag{5}$$

We can then see the effect of the different univariate dimensions by looking at the corresponding elements of $\boldsymbol{w}$. The fact that this is the Fisher's LDA direction is not accidental—we are looking for the direction that maximizes the separation of two groups of data.

3 Analysis of Variance

If we have $L > 2$ algorithms to compare, we test whether they have the same expected performance. In the univariate case, we reduce the confusion matrices to error values and compare them; in the multivariate case, we compare vectors of performance values.

3.1 Univariate Case

Given L populations, we test for

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_L \text{ vs. } H_1 : \mu_r \neq \mu_s \text{ for one pair } r, s$$

Let us say that $e_{ij}, i = 1, \ldots, L, j = 1, \ldots, k$, denotes the error of algorithm i on validation fold j. $e_{i\cdot} = \sum_j e_{ij}/k$ denotes the average error of algorithm i, and $e_{\cdot\cdot} = \sum_i e_{i,\cdot}/L$ denotes the overall average. The univariate ANOVA calculates

$$\begin{aligned} F' &= \frac{MSH}{MSE} = \frac{SSH/(L-1)}{SSE/L(k-1)} \\ &= \frac{(\sum_i e_{i\cdot}^2/k - e_{\cdot\cdot}/Lk)(L-1)}{(\sum_{i,j} e_{ij}^2 - \sum_i e_{i\cdot}^2/k)/L(k-1)} \end{aligned} \tag{6}$$

which, under the null hypothesis, is F distributed with $L-1$ and $L(k-1)$ degrees of freedom. We reject H_0 if $F' > F_{\alpha,L-1,L(k-1)}$.

If ANOVA rejects and we know that there is at least one pair that is significantly different, we can use the pairwise test of Section 2.1 as a post-hoc test on all pairs r, s to check which pair(s) lead(s) to the significant difference in error.

3.2 Multivariate Case

Given L populations, we test for

$$H_0 : \boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \cdots = \boldsymbol{\mu}_L \text{ vs. } H_1 : \boldsymbol{\mu}_r \neq \boldsymbol{\mu}_s \text{ for one pair } r, s.$$

Let us say that $\boldsymbol{x}_{ij}, i = 1, \ldots, L, j = 1, \ldots, k$ denotes the p-dimensional performance vector of algorithm i on validation fold j. The multivariate ANOVA (MANOVA) calculates the two matrices of between- and within-scatter:

$$\mathbf{H} = k \sum_{i=1}^{L} (\overline{\boldsymbol{x}}_{i\cdot} - \overline{\boldsymbol{x}}_{\cdot\cdot})(\overline{\boldsymbol{x}}_{i\cdot} - \overline{\boldsymbol{x}}_{\cdot\cdot})^T$$

$$\mathbf{E} = \sum_{i=1}^{L} \sum_{j=1}^{k} (\boldsymbol{x}_{ij} - \overline{\boldsymbol{x}}_{i\cdot})(\boldsymbol{x}_{ij} - \overline{\boldsymbol{x}}_{i\cdot})^T$$

Then

$$\Lambda' = \frac{|\mathbf{E}|}{|\mathbf{E} + \mathbf{H}|} \tag{7}$$

is *Wilks' Λ* distributed with $p, L - 1, L(k - 1)$ degrees of freedom [4]. We reject H_0 if $\Lambda' \leq \Lambda_{\alpha,p,L-1,L(k-1)}$. Note that rejection is for small values of Λ': If the sample mean vectors are equal, we expect $\mathbf{H}$ to be $\mathbf{0}$ and Λ' to approach 1; as the sample means become more spread, $\mathbf{H}$ becomes "larger" than $\mathbf{E}$ and Λ' approaches 0.

Wilks' Λ can be approximated using χ^2 distribution via the formula

$$\left(\frac{p - n + 1}{2} - m \right) \log \Lambda_{p,m,n} \sim \chi^2_{np} \tag{8}$$

If MANOVA rejects, we can do p separate univariate ANOVA on each of the individual variates as we discussed in Section 3.1, or the difference may be due to some linear combination of the variates: The mean vectors occupy a space whose dimensionality is given by $s = \min(p, L - 1)$; its dimensions are the eigenvectors of $\mathbf{E}^{-1}\mathbf{H}$ and we have

$$\Lambda = \prod_{i=1}^{s} \frac{1}{1 + \lambda_i}$$

where λ_i are the corresponding sorted eigenvalues. The analysis of the eigenvalues and the corresponding variates of the eigenvectors allow us to pinpoint the causes if MANOVA rejects. For example, if $\lambda_1 / \sum_i \lambda_i > 0.9$, there is collinearity, i.e., the means lie on a single discriminant, $z = \boldsymbol{w}^T \boldsymbol{x}$, where $\boldsymbol{w}$ is the eigenvector with the largest eigenvalue λ_1.

We can also do a set of pairwise multivariate tests as we have discussed in Section 2.2 after MANOVA rejects, to see which pairs (or groups) of algorithms have comparable performance vectors.

4 Experiments

4.1 Setup

We use a total of 36 two-class datasets where 27 of them (*artificial, australian, breast, bupa, credit, cylinder, german, haberman, heart, hepatitis, horse, ionosphere, krvskp, magic, mammographic, monks, mushroom, parkinsons, pima,*

polyadenylation, promoters, satellite47, spambase, spect, tictactoe, transfusion, vote) are from the UCI repository [5], three (*ringnorm, titanic, twonorm*) are from the Delve repository [6], and six (*acceptors, ads, dlbcl, donors, musk2, prostatetumor*) are Bioinformatics datasets [7]. We use 10-fold cross-validation and five algorithms: (1) *c45*: C4.5 decision tree. (2) *svm*: Support vector machine (SVM) with a linear kernel [8].(3) *lda*: Linear discriminant classifier. (4) *qda*: Quadratic discriminant classifier. (5) *knn*: k-nearest neighbor with $k = 20$.

4.2 Results

Univariate vs. Multivariate testing. In the first part of our experiments, we compare the univariate k-fold paired t test ($k = 10$) on error which we name UniErr, with our proposed multivariate pairwise test using (tpr, fpr), which we name MultiTF.

Figure 2 shows the example where the univariate test fails to reject and MultiTF rejects the null hypothesis that the two classifiers *lda* and *qda* have the same mean on the *breast* dataset. Figure 2(a) shows the (tpr, fpr) scatter plots of the ten runs each of the two methods and the isoprobability contours of the fitted bivariate Gaussians. We see that LDA has higher fpr whereas QDA has lower tpr, that is, higher false negative rate. We see in Figure 2(b) that the classifiers have comparable overall error histograms: LDA has more false positives, QDA has more false negatives, but overall they have comparable error. We see in Figure 2(c) that the contour plot of the covariance matrix of the paired differences has its mean far from (0,0) and that is why the multivariate test rejects the null hypothesis that the means are the same, whereas in Figure 2(d), we see that histogram of the differences of errors has its mean close to 0 and the univariate test fails to reject the null hypothesis that the means are equal.

MultiTF vs. MultiPR. In the second part of our experiments, we see the effect of different measures on the multivariate test and compare MultiTF with the multivariate test using (precision, recall) that we name MultiPR; this will help us identify which one to use in which context.

Figure 3 shows an example where MultiTF rejects and MultiPR fails to reject the null hypothesis that *c45* and *qda* have the same mean on the *pima* dataset. In Figures 3(a) and (b), the x axes are the same because tpr and recall are the same; the two differ in the y axes and that helps us understand why the two decisions are different. Although with respect to (tpr, fpr), the mean of *c45* and *qda* seem to be close to each other (Fig. 3(a)), their difference is significantly large compared to their standard deviations and this causes a rejection. They are close enough in the (precision, recall) space (Fig. 3(b)) and hence MultiPR does not reject. In calculating precision, we divide by p', and in calculating fpr, we divide by n; here, n is larger than p' and hence, the variance of fpr is smaller, which makes the difference significant.

We can also see this by comparing Figures 3(d) and (e): In (d), we see that (0,0) lies on the outermost contour indicating that the probability that we see a difference as large is small and hence we reject the null hypothesis; in (e), (0,0)

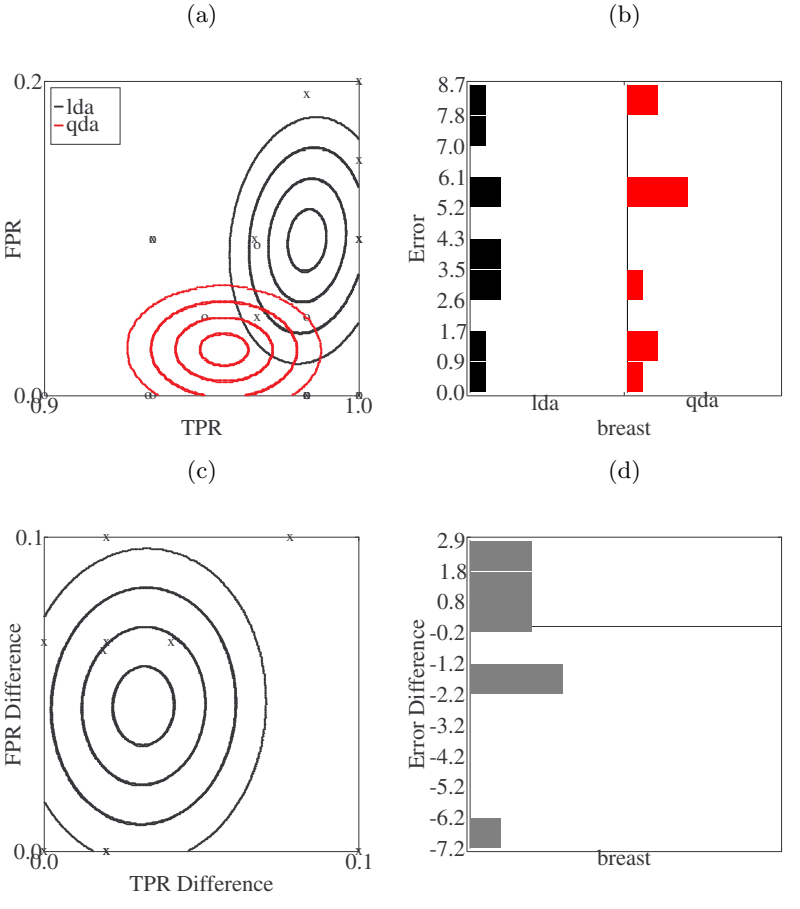


Fig. 2. The example case where the univariate test fails to reject and MultiTF rejects the null hypothesis that *lda* and *qda* have the same mean on the *breast* dataset. (a) shows the isoprobability contour plots of the Gaussians fitted to performance data from two algorithms and (c) shows the distribution of their paired difference; (b) and (d) show the corresponding error histograms and the histogram of paired error differences respectively. Roughly speaking, the multivariate test rejects if the mean of the differences is far from (0,0), compared to the scale of the covariance matrix of differences; just as the univariate test rejects if the mean of the differences is far from 0, compared to the standard deviation of differences.

is close enough to the center of the contours and the probability that we see such a difference is not small and hence we do not reject.

If the univariate post-hoc tests are performed, we see that the algorithms are significantly different in terms of fpr with a p-value of 0.006. The corresponding elements of $\boldsymbol{w}$ (equation 5) are (tpr : 0.499, fpr : -25.898) and (precision : 2.014, recall : -4.374), which shows that fpr is the important one.

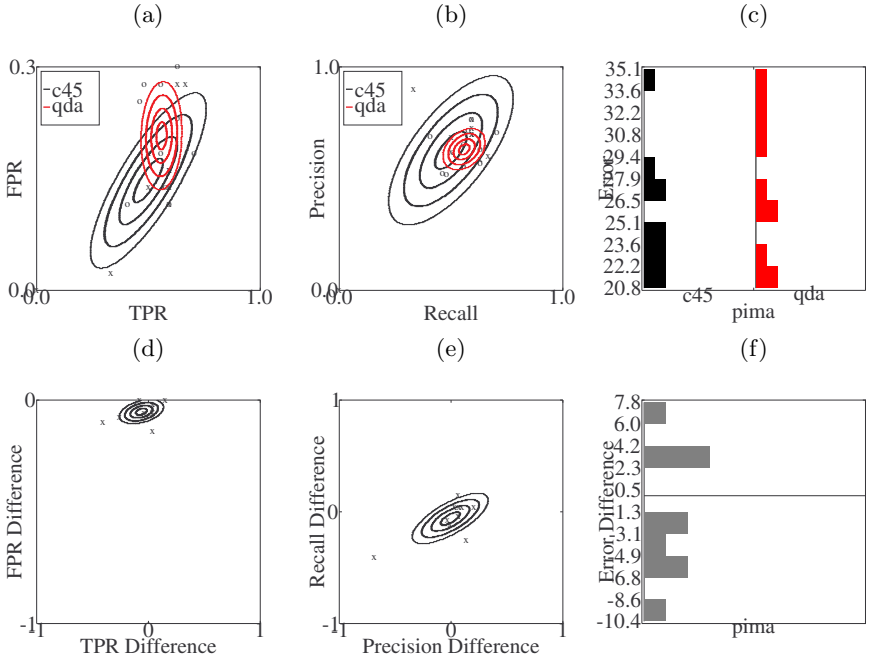


Fig. 3. An example case where MultiTF rejects and MultiPR fails to reject the null hypothesis that the two classifiers, *c45* and *qda*, have the same mean on the *pima* dataset. (a) and (d) show the isoprobability contour plots of the fitted Gaussians and of the difference with respect to (tpr, fpr); (b) and (e) show the same with respect to (precision, recall); (c) and (f) show the corresponding histogram of the error rates and the differences in the error rates.

The error distributions of the algorithms are also similar to each other and the univariate test also fails to reject the null hypothesis that the error rates of those algorithms are equal (see Figs. 3(c) and (f)).

(Precision, recall) and (tpr, fpr) metric pairs have different application areas. In (precision, recall), we are basically interested in how well we classify the positive examples, whereas in (tpr, fpr), in trying to minimize fpr, we also want to increase the true negatives. To show the difference between them, we did two experiments: In Figure 4(a), we simply add more and more true negatives to a classifier. In such a case, we see that this has no effect on precision and recall, but decreases fpr. When compared with the classifier without any additional true negatives, MultiPR does not reject but MultiTF starts rejecting after a point.

It is known that (precision, recall) is sensitive to class skewness [9], whereas (tpr, fpr) is not. In Figure 4(b)), we slowly change the ratio p/n, and we see that because precision uses values from both rows, it changes; however (tpr, fpr) do not change since they use values from only one row. Compared with the classifier with the original ratio, MultiTF does not reject (because the rates do not change), but MultiPR starts rejecting after a point.

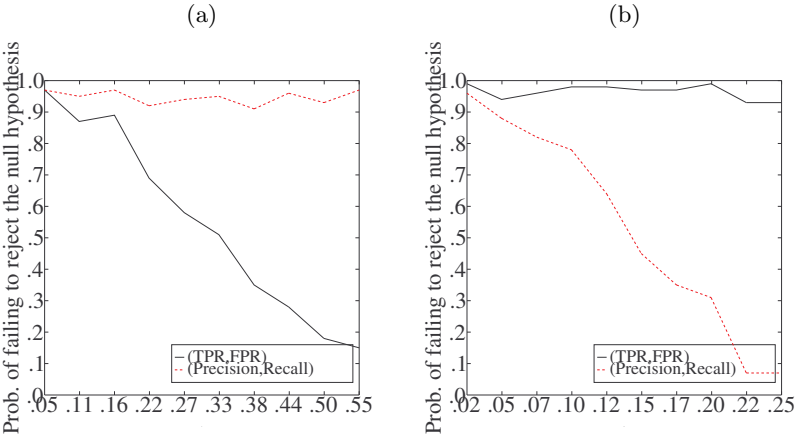


Fig. 4. In (a), when we add more and more true negatives ($tn \leftarrow tn\ (1+\lambda)$), precision and recall do not change, but fpr ($=fp\ /\ (n+\lambda\ tn)$) decreases and MultiTF test starts rejecting the null hypothesis. In (b), we change the ratio $\frac{p}{n} = \frac{(tp+fn)(1-\alpha)}{(fp+tn)(1+\alpha)}$ while keeping tpr and fpr the same ($tp \leftarrow tp\ (1-\alpha)$, $fn \leftarrow fn\ (1-\alpha)$, $fp \leftarrow fp\ (1+\alpha)$, $tn \leftarrow tn\ (1+\alpha)$), we see that precision changes and MultiPR starts rejecting. Plotted values are proportions of failures to reject in 100 independent runs.

If we are doing an information retrieval task with a query such as, "Find me all images of tigers," adding additional non-tiger images to the database does not have any effect on our measure of performance (as long as we have no difficulty in recognizing them as non-tigers and do not retrieve them), and hence we use precision and recall. If we want to differentiate between two types of targets, for example, cars and tanks, our accuracy on these different targets is important, and we use tpr and fpr.

Comparison of multiple algorithms. In the third part of our experiments, we use the univariate and multivariate tests to compare $L > 2$ classification algorithms. For the univariate case, if ANOVA rejects, we can do $L(L-1)/2$ pairwise univariate tests to find difference between pairs and also cliques, i.e., subsets of algorithms in which all pairwise tests fail to reject.

In the case of a univariate test, we can also write down an order by comparing the means. For this, we sort the algorithms in terms of average error in ascending order and then try to find groups where there is no statistically significant difference between the smallest and largest means in the group, which we check by applying a pairwise univariate test to these two at the ends. If this the case, we underline the group. We first try all five, if there is a difference between the first and the fifth, we try the two groups of four leaving out the two extremes, and so on.

If MANOVA rejects, similarly, we can do the pairwise multivariate tests and find cliques. We can also do univariate tests on the dimensions separately and

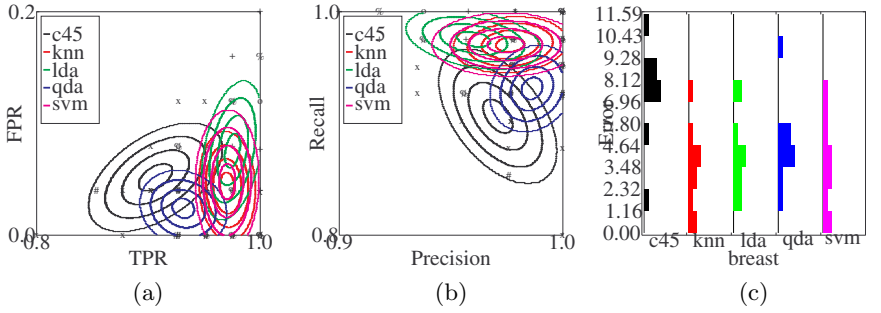


Fig. 5. Comparison of five algorithms on *breast*. (a) and (b) show the isoprobability contour plots of the fitted bivariate Gaussians with respect to (tpr, fpr) and (precision, recall) respectively; (c) shows the corresponding histogram of the error rates.

Table 2. Tabular representation of post-hoc univariate and MultiTF/MultiPR test results on *breast* dataset. **1** stands for a failure to reject the null hypothesis.

	c45	*lda*	*qda*	*svm*	*knn*
c45	0	0	0	0	0
lda	0		**1**	**1**	**1**
qda	0	**1**		**1**	**1**
svm	0	**1**	**1**		**1**
knn	0	**1**	**1**	**1**	

	c45	*lda*	*qda*	*svm*	*knn*
c45	0	0	0	0	0
lda	0		0	**1**	**1**
qda	0	0		0	0
svm	0	**1**	0		**1**
knn	0	**1**	0	**1**	

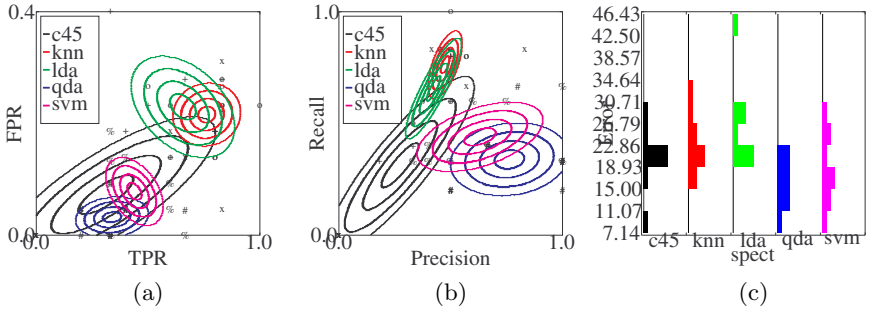
try to find orderings, as discussed above for error. For example, if MANOVA on (precision, recall) on five algorithms reject, we can try to find groups and orderings in terms of precision and recall separately.

Figure 5 shows the first example case on *breast* dataset. Both ANOVA and MANOVA reject the null hypothesis. According to post-hoc test results, the univariate test finds a single clique of four algorithms (*knn*, *lda*, *qda*, *svm*). On the other hand, both multivariate post-hoc tests (MultiTF and MultiPR) find a single clique of three algorithms (*knn*, *lda*, *svm*). Table 2 shows the results of all pairwise tests between five algorithms.

The univariate orderings found are as follows:

error	*knn svm lda qda c45*
tpr, recall	*lda knn svm qda c45*
fpr	*qda knn svm c45 lda*
precision	*qda knn svm c45 lda*

The clique found by multivariate tests (*knn*, *lda*, *svm*) appears as a single group with respect to tpr and recall. Although (*knn*, *svm*) appear together, *lda* is separate from that group when the criterion is fpr or precision. We see that these different measures are able to detect differences that error cannot, and that the differences vary depending on what performance measure we concentrate on.



(a) (b) (c)

Fig. 6. Comparison of five algorithms on *spect*. (a) and (b) show the isoprobability contour plots of the fitted bivariate Gaussians with respect to (tpr, fpr) and (precision, recall) respectively; (c) shows the corresponding histogram of the error rates.

Table 3. Tabular representation of post-hoc MultiTF and MultiPR test results on *spect* dataset. **1** stands for failing to reject the null hypothesis.

	c45	*lda*	*qda*	*svm*	*knn*
c45		0	**1**	**1**	0
lda	0		0	0	**1**
qda	**1**	0		**1**	0
svm	**1**	0	**1**		0
knn	0	**1**	0	0	

	c45	*lda*	*qda*	*svm*	*knn*
c45		0	0	0	0
lda	0		0	0	**1**
qda	0	0		**1**	0
svm	0	0	**1**		0
knn	0	**1**	0	0	

Figure 6 shows the second example case where we compare all algorithms on *spect*. Again, both ANOVA and MANOVA reject the null hypothesis. According to the post-hoc tests, the univariate test finds five different cliques (one clique of three and four cliques of two algorithms): (*c45*, *qda*, *svm*), (*knn c45*), (*lda*, *c45*), (*lda*, *knn*), (*svm*, *knn*). On this dataset, the decisions of the two multivariate tests, MultiTF and MultiPR, are different from each other. MultiTF finds two cliques: (*c45*, *qda*, *svm*) and (*lda*, *knn*), whereas MultiPR finds the same cliques except *c45* is missing in one clique: (*qda*, *svm*) and (*lda*, *knn*). Table 3 shows the results of the multivariate pairwise tests between five algorithms.

The univariate ordering of the five classifiers are as follows:

error	*qda svm c45 knn lda*
tpr, recall	*knn lda svm qda c45*
fpr	*qda c45 svm knn lda*
precision	*qda svm knn lda c45*

The first clique found by MultiTF (*c45*, *qda*, *svm*) appears as a single group with respect to both tpr and fpr, whereas the second clique (*lda*, *knn*) form a group only with respect to fpr. Similarly, the first clique found by MultiPR (*qda*, *svm*) appears as a single group with respect to recall and precision, whereas the second clique (*lda*, *knn*) form a group only with respect to precision.

Using the full 2×2 confusion matrix. Instead of using (tpr, fpr) or (precision, recall), one can also use the full 2×2 confusion matrix using the same multivariate test in four dimensions. Note however that though the matrix contains four numbers, because p and n are fixed, the degree of freedom is two and that going to four dimensions is unnecessary. In our pairwise comparison experiments, we see that in 2582 cases out of 2740, the rank of the 2×2 confusion matrix $\mathbf{S}_d$ is indeed 2. Only in 98 cases, the rank is 1: This case occurs if the ratio tp / tn is the same for all folds, and in 60 cases, the rank is 4: This case occurs if the number of positive and/or negative instances is not exactly divisible by k, resulting in a difference between the positive and/or negative instances going from one fold to another.

It can be shown that when we use the 2×2 confusion matrix, the test statistic calculated will be the same as that of MultiTF. The other values fn and tn are fixed because we have ($p = tp + fn$) and ($n = fp + tn$) and they do not change going from one fold to another due to stratification, reducing the dimensionality to two, and it can be shown that MultiTF uses scaled versions of the counts used by Multi 2×2 but both return the same value. As explained above, there are cases when the stratification is not exact, but such cases are rare and do not affect the overall result.

5 Conclusions

In this paper, we propose to use multivariate tests to compare the performances of classification algorithms. Doing this, we can consider entries in the confusion matrix separately without needing to sum them up in a cumulative measure such as error or accuracy, which may hide certain differences in the behavior of the algorithms. Though multivariate pairwise tests and multivariate ANOVA have been known in the statistical literature, to the best of our knowledge, their use in performance comparison of machine learning algorithms is new.

There are a number of advantages to testing p variables multivariately rather than p separate univariate testing [4], as has also been shown in our experimental results above: (1) The use of p univariate tests inflates the type I error rate, unless we do some sort of correction (which in turn decreases power). (2) The univariate tests ignore correlations between variables, whereas the multivariate test uses the covariance information. (3) The multivariate test has higher power: Sometimes the p univariate tests may fail to detect a difference whereas the multivariate difference may be significant. (4) The multivariate test (pairwise test or MANOVA) constructs linear combinations of variables that reveals how the variables unite to reject the hypothesis.

The use of k-fold cross-validation to obtain k set of performance values comes with a caveat. Because all k training/validation sets are resampled from the same set, they overlap, and these k set of measurements are not really independent. This is true both for the univariate t test [1] and the multivariate test. Nadeu and Bengio (2003) [10] and Bouckaert and Frank (2004) [11] discuss a variance-correction term. We note that the resampling procedure used to generate the k

data folds is orthogonal to the test which uses these results and that our proposed multivariate test can be used with any improved resampling procedure.

We calculate (tpr, fpr) or (precision, recall) values for a specific threshold value. To have an overall comparison, for example, we can use s different thresholds (as done in a ROC curve) and calculate a pair for each value and get an overall $2s$ dimensional vector and again use the multivariate test. This is an interesting research direction. In this paper, we discuss how two or more algorithms can be compared on a single dataset. Demsar (2006) [12] discusses the comparison of algorithms over multiple datasets and an interesting future direction will be to extend our proposed multivariate test for this.

Acknowledgments

This work has been supported by TÜBİTAK 109E186.

References

1. Dietterich, T.G.: Approximate statistical tests for comparing supervised classification learning classifiers. Neural Computation 10, 1895–1923 (1998)
2. Caruana, R., Niculescu-Mizil, A., Crew, G., Ksikes, A.: Ensemble selection from libraries of models. In: Proceedings of the International Conference on Machine Learning, ICML 2004, pp. 137–144 (2004)
3. Seliya, N., Khoshgoftaar, T.M., Hulse, J.V.: Aggregating performance metrics for classifier evaluation. In: Proceedings of the 10th IEEE International Conference on Information Reuse and Integration (2009)
4. Rencher, A.C.: Methods of Multivariate Analysis. Wiley and Sons, New York (1995)
5. Blake, C., Merz, C.: UCI repository of machine learning databases (2000)
6. Hinton, G.H.: Delve project, data for evaluating learning in valid experiments (1996)
7. Statnikov, A., Aliferis, C., Tsamardinos, I., Hardin, D., Levy, S.: A comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 21, 631–643 (2005)
8. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines (2001)
9. Davis, J., Goadrich, M.: The relationship between precision-recall and roc curves. In: Proceedings of the 23rd International Conference on Machine Learning, vol. 148, pp. 233–240 (2006)
10. Nadeau, C., Bengio, Y.: Inference for the generalization error. Machine Learning 52, 239–281 (2003)
11. Bouckaert, R., Frank, E.: Evaluating the replicability of significance tests for comparing learning algorithms. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 3–12. Springer, Heidelberg (2004)
12. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. Journal of Machine Learning Research 7, 1–30 (2006)